

# ALGORITMIA Y COMPLEJIDAD

## LABORATORIO - TEMA 5

### EJERCICIO 1

Se tienen  $N$  elementos distintos almacenados en una estructura de acceso directo, por ejemplo, un vector  $[1, 2, 3, 4, 5]$ , o una cadena "abcdefg".

Diseñar un algoritmo que use Backtracking para obtener todas las formas distintas de colocar los elementos, es decir, el algoritmo debe calcular y devolver todas las permutaciones de los  $N$  elementos.

### EJERCICIO 2

Resolver el problema anterior considerando la posibilidad de que los elementos se repitan entre sí, por ejemplo, el vector  $[1, 2, 3, 1]$  o la cadena "acabada".

### EJERCICIO 3

Se tiene un número de  $N$  cifras almacenado en una cadena de texto; por ejemplo, la cadena "1151451".

Diseñar un algoritmo que, mediante técnicas de Backtracking, encuentre todos los números distintos de  $N$  cifras que puedan formarse con los dígitos de la cadena, sin alterar su orden relativo dentro de la misma.

Por ejemplo, si  $N = 4$ , son números válidos 1151, 1511 y 1541, pero no 4551 o 5411, ya que, aunque pueden formarse con los dígitos de la cadena, implican una reordenación.

### EJERCICIO 4

Se dispone de un tablero  $M$  de tamaño  $F \times C$  (donde  $F$  es la cantidad de filas y  $C$  la cantidad de columnas) y se pone en una casilla inicial  $(x, y)$  un caballo de ajedrez.

El objetivo es encontrar, si es posible, la forma en la que el caballo debe moverse para recorrer todo el tablero, de manera que cada casilla se utilice una única vez en el recorrido. El tablero  $8 \times 8$  siempre tiene solución independientemente de dónde comience el caballo. El caballo puede terminar en cualquier posición del tablero.

El caballo de ajedrez tiene ocho posibles movimientos (suponiendo que no se sale del tablero).

Un movimiento entre las casillas  $M_{ij}$  y  $M_{pq}$  es válido solamente si se cumple alguna de las siguientes condiciones:

- $|p - i| = 1$  y  $|q - j| = 2$
- $|p - i| = 2$  y  $|q - j| = 1$

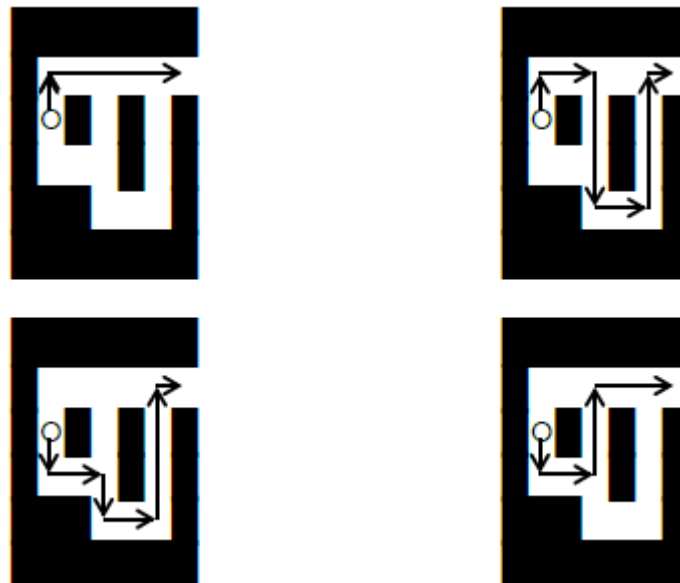
Es decir, una coordenada cambia dos unidades y la otra una única unidad.

### EJERCICIO 5

Se dispone de una tabla  $[1..n, 1..m]$  con valores lógicos que representan un laberinto. El valor True indica la existencia de una pared (no se puede atravesar), mientras que el valor False representa un camino.

Para moverse por el laberinto, a partir de una casilla dada, podemos desplazarnos en cualquier dirección con un movimiento horizontal o vertical, pero solo a una casilla vacía (que contenga False).

Los bordes de la tabla están completamente a True excepto una casilla, que es la salida del laberinto.



Diseñar un algoritmo de Backtracking, que encuentre todos los caminos posibles que llevan a la salida desde una casilla inicial, siempre que sea posible salir del laberinto.

## EJERCICIO 6

Se tiene la tabla M de sustitución que aparece a continuación:

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>
<b>a</b>	b	b	a	d
<b>b</b>	c	a	d	a
<b>c</b>	b	a	c	c
<b>d</b>	d	c	d	b

En una cadena cualquiera, dos caracteres consecutivos se pueden sustituir por el valor que aparece en la tabla, utilizando el primer carácter como fila y el segundo como columna. Por ejemplo, se puede cambiar la secuencia “ca” por una “b”, ya que  $M[c][a]=b$ .

Implementar un algoritmo Backtracking que, a partir de una cadena de texto, sea capaz de encontrar la forma de realizar las sustituciones para reducir la cadena a un solo carácter dado, si es posible.

Ejemplo: Dada la cadena “acabada” y el carácter “d”, una posible forma de sustitución es la siguiente (las secuencias que se sustituyen se marcan para mayor claridad): acabada  $\rightarrow$  acada  $\rightarrow$  abcda  $\rightarrow$  abcd  $\rightarrow$  bcd  $\rightarrow$  bc  $\rightarrow$  d.

**Entregables:** un ejercicio a elegir entre los problemas 3 y 4 y el ejercicio 6.